



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ 421 – Προγραμματισμός Συστημάτων

ΑΣΚΗΣΗ 3 – Υλοποίηση Προγράμματος Αναζήτησης μέσω του Συστήματος Αρχείων (find)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ
Υπεύθυνος Εργαστηρίου: Πάυλος Αντωνίου

Ημερομηνία Ανάθεσης: Τετάρτη 30/10/2024
Ημερομηνία Παράδοσης: Τετάρτη 13/11/2024 @ 13:00 (14 μέρες)

I. Στόχος Άσκησης

Στόχος αυτής της άσκησης είναι η εξοικείωση με προχωρημένες τεχνικές προγραμματισμού στο περιβάλλον UNIX. Συγκεκριμένα, σε αυτή την άσκηση θα έχετε την δυνατότητα να κάνετε πρακτική εξάσκηση των εννοιών που έχετε διδαχθεί στις διαλέξεις όπου είδαμε τη *Δομή Αρχείων, το Χαμηλού Επιπέδου I/O, Διαχείριση μετα-πληροφοριών αρχείων <sys/stat.h>, Διαχείριση αρχείων με κλήσεις συστήματος και Προσπέλαση και Επεξεργασία καταλόγων (dirent.h) μέσω κλήσεων βιβλιοθήκης*. Η υλοποίηση της άσκησης προϋποθέτει ότι έχετε κατανοήσει την δομή του συστήματος αρχείων στο UNIX καθώς επίσης την λειτουργία των I-nodes.

Η άσκηση περιλαμβάνει την υλοποίηση των λειτουργιών της εντολής GNU find, χωρίς την κλήση της οικογένειας εντολών exec. Οι λειτουργίες του προγράμματος σας και το αναμενόμενο αποτέλεσμα περιγράφονται αναλυτικότερα στην συνέχεια.

II. Περιγραφή της εντολής find

Η εντολή find **αναζητά** μέσα σε **ένα ή περισσότερα δέντρα καταλόγων** ενός συστήματος αρχείων, **εντοπίζει αρχεία** που ταιριάζουν με καθορισμένα (από το χρήστη) κριτήρια, και **εφαρμόζει σε αυτά μια πράξη** (action) που καθορίζεται από το χρήστη.

Τα πιθανά κριτήρια αναζήτησης περιλαμβάνουν ένα σχέδιο (pattern) που ταιριάζει με το όνομα του αρχείου/ων ή ένα εύρος χρόνου για να ταιριάζει με το χρόνο τροποποίησης ή το χρόνος πρόσβασης του αρχείου/ων.

Σύνταξη εντολής

```
find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]
```

Οι τρεις επιλογές (-H, -L, -P) προκαθορίζουν τον τρόπο που η εντολή find θα αντιμετωπίζει συμβολικούς συνδέσμους (symbolic links). Η χρήση τους είναι προαιρετική. Η προεπιλεγμένη (default) συμπεριφορά είναι να μην ακολουθηθούν οι συμβολικοί σύνδεσμοι. Αυτό μπορεί να ορίζεται ρητώς με την επιλογή -P. Η επιλογή -L θα αναγκάσει την εντολή find να ακολουθηθεί συμβολικούς συνδέσμους. Αυτό πρακτικά σημαίνει, αν υπάρχει κάποιος σύνδεσμος που παραπέμπει σε κάποιο άλλο κατάλογο (έστω X/), η find θα πρέπει να συνεχίσει την αναζήτηση μέσα στο φάκελο X/. Η επιλογή -H θα αναγκάσει την εντολή find να ακολουθηθεί *μόνο* συμβολικούς συνδέσμους κατά την επεξεργασία των arguments της γραμμής εντολών (π.χ., "find -H

symbolic-link-file.txt aa/ " θα ακολουθήσει και το symbolic-link-file.txt). Παρακαλώ δείτε το εγχειρίδιο "man find" για αναλυτικότερες οδηγίες.

Η επιλογές -H, -L, -P, -D debugopts και -Olevel δεν χρειάζεται να υλοποιηθούν στην παρούσα άσκηση εφόσον αφορούν θέματα αποσφαλμάτωσης και βελτιστοποίησης της εκτέλεσης.

Στη συνέχεια, μία διαδρομή (path) προηγείται της έκφρασης (expression). Αν δεν δίνεται κάποια διαδρομή, η προεπιλεγμένη συμπεριφορά της find είναι να ξεκινήσει η αναζήτηση από τον τρέχον κατάλογο (.).

Η λειτουργία της εντολής find είναι αναδρομική, καθώς αρχίζει την αναζήτηση μέσα στον κατάλογο της δοθείσας διαδρομής (path) και συνεχίζει σε όλους τους υποκαταλόγους. Ακολουθώντας, πάνω σε κάθε επεξεργαζόμενο κατάλογο γίνεται η αποτίμηση των expression options, π.χ., σε πόσο βάθος να εκταθεί η αναδρομή, ποιες ημερομηνίες, κτλ. όπως αναλύεται στην επόμενη ενότητα. Για παράδειγμα,

```
find ~ -maxdepth 3 -type f -name "core*" -print
```

θα αναζητήσει μόνο αρχεία (-type f) των οποίων το όνομα ξεκινά από core (-name "core*") που βρίσκονται στο HOME directory (~) του χρήστη και σε όλους τους υποκαταλόγους που είναι μέσα στο ~ μέχρι και βάθους 3 (δηλαδή ο κατάλογος ~ (βάθος 1), οι υποκατάλογοι του ~ (βάθος 2), και οι υποκατάλογοι των υποκαταλόγων του ~ (βάθος 3)) και θα εκτυπώσει (-print) μόνο όσα ταιριάζουν με τα κριτήρια. Στην εντολή αυτή, το δέντρο καταλόγων προς αναζήτηση έχει σαν ρίζα το HOME directory (~), τα κριτήρια της αναζήτησης είναι τα -maxdepth 3, -name "core*" και -type f, και η πράξη που εφαρμόζεται πάνω στα αρχεία που ταιριάζουν με τα κριτήρια είναι η εκτύπωση (-print). Αξίζει να σημειωθεί ότι προεπιλεγμένη (default) πράξη της εντολής find είναι η -print και ως εκ τούτου η ρητή η δήλωσή της μπορεί να παραληφθεί.

III. Ζητούμενα Άσκησης

Στα πλαίσια της εργασίας αυτής πρέπει να υποστηρίξετε τις ακόλουθες expression options:

```
--type c
File is of type c:
b      block (buffered) special
c      character (unbuffered) special
d      directory
p      named pipe (FIFO)
f      regular file
l      symbolic link; this is never true if the -L option or the -follow
      option is in effect, unless the symbolic link is broken.  If you
      want to search for symbolic links when -L is in effect,
      use -xtype.
s      socket
D      door (Solaris)

--uid n
File's numeric user ID is n.

--user uname
File is owned by user uname (numeric user ID allowed).

--gid n
File's numeric group ID is n.
```

--group gname
File belongs to group gname (numeric group ID allowed).

--name pattern
Base of file name (the path with the leading directories removed) matches shell pattern pattern. The metacharacters ('*', '?', and '[') match a '.' at the start of the base name. The filename matching is performed with the use of the fnmatch(3) library function. Don't forget to enclose the pattern in quotes in order to protect it from expansion by the shell.

--newer file
File was modified more recently than file. If file is a symbolic link and the -H option or the -L option is in effect, the modification time of the file it points to is always used.

--iname pattern
Like -name, but the match is case insensitive. For example, the patterns 'fo*' and 'F??' match the file names 'Foo', 'FOO', 'foo', 'fOo', etc. In these patterns, unlike filename expansion by the shell, an initial '.' can be matched by '*'. That is, find -name *bar will match the file '.foo-bar'. Please note that you should quote patterns as a matter of course, otherwise the shell will expand any wildcard characters in them.

--inum n
File has inode number n. It is normally easier to use the -samefile test instead.

--depth
Process each directory's contents before the directory itself (i.e., Depth-First-Search using a Stack).

--help, --help
Print a summary of the command-line usage of find and exit.

--maxdepth levels
Descend at most levels (a non-negative integer) levels of directories below the command line arguments. -maxdepth 0 means only apply the tests and actions to the command line arguments.

--mindepth levels
Do not apply any tests or actions at levels less than levels (a non-negative integer). -mindepth 1 means process all files except the command line arguments.

--mount
Don't descend directories on other filesystems (please read local filesystem prefixes from /etc/fstab). An alternate name for -xdev, for compatibility with some other versions of find.

--version, --version
Print the find version number and exit.

IV. Βοηθητικές Συναρτήσεις & Οδηγίες

Για την διεκπεραίωση αυτής της άσκησης θα χρειαστείτε ενδεχομένως κάποιες συναρτήσεις τις οποίες δεν είδαμε στο μάθημα. Συγκεκριμένα οι ακόλουθες συναρτήσεις (Κεφάλαιο 6 – Stevens & Rago) μπορεί να είναι χρήσιμες (για περαιτέρω συναρτήσεις ανατρέξτε το βιβλίο και το man):

```
#include <pwd.h>
struct passwd *getpwuid(uid_t uid);
Returns: Pointer to struct passwd if OK, NULL on error
```

```
#include <grp.h>
struct group *getgrgid(gid_t gid);
Returns: Pointer to struct group if OK, NULL on error
```

α) struct passwd *getpwuid(uid_t uid);

Αντιστοιχεί το uid το οποίο περιέχετε στο i-node ενός αρχείου σε μια πιο εύληπτη μορφή (π.χ. αντί 1240 δίδει dzeina). Η εντολή αυτή ουσιαστικά ψάχνει μέσα στο /etc/passwd αρχείο. Η struct passwd περιέχει το πεδίο pw_name το οποίο είναι τύπου (char *) και το οποίο περιέχει το όνομα του χρήστη.

β) struct group *getgrgid(gid_t gid);

Αντιστοιχεί το gid το οποίο περιέχετε στο i-node ενός αρχείου σε μια πιο εύληπτη μορφή (π.χ. αντί 243 δίδει visiting). Η εντολή αυτή ουσιαστικά ψάχνει μέσα στο /etc/group αρχείο. Η struct group περιέχει το πεδίο gr_name το οποίο είναι τύπου (char *) και το οποίο περιέχει το όνομα του χρήστη.

γ) Μελετήσετε το “**man stat**” για να κατανοήσετε όλες τις σταθερές της βιβλιοθήκης <sys/stat.h>

δ) Σημαντική παρατήρηση από την εντολή find που δόθηκε πιο πάνω αποτελεί η ύπαρξη της κανονικής έκφρασης “core*” δίπλα από το κριτήριο -name. Η εντολή find υποστηρίζει κανονικές εκφράσεις (regular expressions) για την εύρεση ονομάτων αρχείων και ως εκ τούτου θα βρείτε χρήσιμη τη χρήση της βιβλιοθήκης **fnmatch.h** ξεκινώντας από το κέλυφος με χρήση της **man fnmatch.h** (η οποία είναι εξειδικευμένη για κανονικές εκφράσεις σε ονόματα αρχείων, σε αντίθεση με την regex.h η οποία είναι ευρύτερη για κανονικές εκφράσεις).

ε) Για την επεξεργασία των επιλογών του προγράμματος σας (command line arguments) θα ήταν καλό να κάνετε χρήση της βιβλιοθήκης **getopt_long** (για περισσότερες πληροφορίες “man 3 getopt_long”, όπου 3 δηλώνει “C library routines”).

Σημειώστε ότι η εφαρμογή σας πρέπει να υποστηρίζει οποιουδήποτε συνδυασμούς των πιο πάνω ορισμάτων και να παράγει το αναμενόμενο αποτέλεσμα. Η αναμενόμενη λειτουργία κάθε συνδυασμού ορισμάτων μπορεί να βρεθεί με χρήση της ίδιας της **find**.

V. Γενικοί Κανόνες

1. Το πρόγραμμα σας πρέπει να χρησιμοποιεί τεχνικές δομημένου προγραμματισμού με την χρήση συναρτήσεων, πολλαπλών αρχείων για καλύτερη δομή του πηγαίου κώδικα, Makefile, διαχείριση λαθών συστήματος με την perror κτλ.;
2. Το πρόγραμμα σας πρέπει να ελαχιστοποιεί την χρήση πόρων του συστήματος (μνήμης, επεξεργαστή, στίβας προγράμματος, κτλ);
3. Το πρόγραμμα πρέπει να μειώνει όσο το δυνατό περισσότερο τον χρόνο διεκπεραίωσης της εντολής.

Καλή Επιτυχία !